UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/807,529	03/23/2004	Hieu Trung Tran		3430

7590 01/29/2007

HIEU TRUNG TRAN / VIOSOFT CORP.
SUITE 203A
2959 S. WINCHESTER BLVD
CAMPBELL, CA 95008

EXAMINER	
PUENTE, EMERSON C	
ART UNIT	**PAPER NUMBER**
2113	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/29/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/807,529	TRAN, HIEU TRUNG	
	Examiner	Art Unit	
	Emerson C. Puente	2113	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 March 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-45* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *23 March 2004* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claims 1-45 have been examined.

This action is made **Non-Final**.

Drawings

The drawings are objected to because of the following informalities:

Please remove the title, inventor name, and phone number from the top of each figure, as they are unnecessary.

Appropriate correction is required.

Specification

The abstract of the disclosure is objected to because the abstract is not in a single paragraph of 150 words or less. Correction is required. See MPEP § 608.01(b).

The disclosure is objected to because of the following informalities:

Please change "wais"(see page 12, paragraph 34, line 1) to "waits"..

Appropriate correction is required.

Claim Objections

Claims 7, 22, and 37 objected to because of the following informalities:

Regarding claims 7,22, and 37, please remove either "the" or "said" from the limitation "the said proxy"(see second to last line of claims) as only one is needed.

Appropriate correction is required.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 37-45 are rejected under 35 U.S.C. 101 because claimed invention is directed to non-statutory subject matter. A program product, without the computer-readable medium needed to realize the computer program's functionality, constitutes nonstatutory subject matter. See MPEP § 2106. Examiner suggests amending "A program product ..." to "A program product stored on a computer readable medium ... ".

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-45 rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The claims are indefinite as they are replete with lack of antecedent problems. Antecedent problems arise when a claim contains words or phrases whose meaning is unclear. As stated in the MPEP, the lack of clarity could arise where a claim refers to "said lever" or "the lever," where the claim contains no earlier recitation or limitation of a lever and where it would be unclear as to what element the limitation was making reference. Similarly, if two different

levers are recited earlier in the claim, the recitation of "said lever" in the same or subsequent claim would be unclear where it is uncertain which of the two levers was intended. A claim which refers to "said aluminum lever," but recites only "a lever" earlier in the claim, is indefinite because it is uncertain as to the lever to which reference is made (see MPEP 2173.05(e)).

Claims 1-45 should be revised carefully in order to comply with 35 U.S.C. 112, second paragraph. Examples of some lack of antecedent basis problems include:

"the target computer" in line 3 of claim 1. It is unclear whether or not the applicant intended to make reference to the "remote computer"(see line 1) or another computer. If the target computer is in reference to "a remote computer", examiner suggesting amending "the target computer" to "the remote computer". If the target computer is different from the remote computer, examiner suggest amending "the target computer" to "a target computer".

"the processing of debug traps" in line 7-8 of claim 1. Examiner suggest amending "the processing of debug traps" to "processing of debug traps".

"said replaced code and data" in lines 8-9 of claim 1. It is unclear whether or not the applicant intended to make reference to "OS kernel code and data"(see line 6). If so, examiner suggests amending "said replaced code and data" to "said OS kernel code and data".

"the target" in line 11 of claim 1. It is unclear whether or not the applicant intended to make reference to the "target computer"(see line 3). If so, examiner suggests amending "the target" to "the target computer". Examiner further notes if applicant intended "the target computer" (see line 3 of claim) to refer to the "remote computer"(see line 1 of claim), as indicated above, "the target"(see line 11) would need to be change "to remote computer" so as to reference the remote computer.

"replaced OS kernel and data" in last line of claim 1. It is unclear whether or not the applicant intended to make reference to "OS kernel code and data"(see line 6). If so, examiner suggests amending "replaced OS kernel and data" to "said OS kernel code and data".

"the OS loadable module loading system call" in line 1 of claim 7. Examiner suggests amending to "the OS loadable module loading system call".

"the target system" in line 2 of claim 7. Examiner suggests amending to "a target system".

"the OS syscall table" in line 3 of claim 7. Examiner suggests amending to "a OS syscall table".

"the OS module loading system call" in line 4 of claim 7. It is unclear whether or not applicant intended to make reference to the "OS loadable module loading system call" (see line 1). If so, examiner suggests amending to "the OS loadable module loading system call".

"the debug agent memory image" in line 6 of claim 7. Examiner suggests amending to "a debug agent memory image".

"the loading utility program" in lines 8-9 of claim 7. Examiner suggests amending to "a loading utility program".

"the steps comprising" in line 9 of claim 7. Examiner suggests amending to "steps comprising".

"said replacement" in line 10 of claim 7. It is unclear whether or not applicant intended to make reference to "replacing said entry (see line 5)". If so, examiner suggests amending to "said replacing said entry".

"said module" in line 11 of claim 7. It is unclear whether or not applicant intended to make reference to the "OS loadable module"(see line 4 and 7). If so, examiner suggests amending to "the OS loadable module".

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 7 and 37 are rejected under 35 U.S.C. 102(b) as being anticipated by US Patent No. 6,189,140 of Madduri.

In regards to claims 7 and 37, Madduri discloses a method of and computer program product for intercepting the OS loadable module loading system call, comprising:

running a debug agent on the target system. Madduri discloses entry into a hardware debug mode, indicating a hardware debugger or agent (see column 16 lines 16-23).

saving a entry in the OS syscall table pointing to a sys_init_module function, which services the OS module loading system call. Madduri discloses setting a bit that causes entry into a debug mode or software interrupt (OS module loading system call) (see column 16 lines 16-17). In order to toggle control back and forth between software interrupt and debug mode, an entry must be saved.

replacing said entry with a pointer to a proxy sys_init_module function residing in the debug agent memory image. Madduri discloses setting a bit that causes entry into a debug mode that otherwise cause a software interrupt (see column 16 lines 16-17).

responsive to system call to load the loadable module by the loading utility program, executing the steps comprising:

invoking said proxy sys_int_module function via said replacement. Madduri discloses setting a bit, indicating replacement, that causes entry into a debug mode (see column 16 lines 16-17).

responsive to determining that said module has been selected for debugging, initiating debugging of the loadable module by the said proxy sys_init_module function. Madduri discloses entry into a debug mode for all debug traps/fault of the processor core (see column 16 lines 16-17).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 16, 22, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Madduri in view of US Patent Application Publication No. 2003/0074650 of Akgul et al. referred hereinafter "Akgul".

In regards to claims 1 and 31, Madduri discloses a method of debugging a remote computer and a program product, comprising:

running a debugger on a host computer (see figure 1 item 111,112 and column 4 lines 46-50)

running an operating system on the target computer (see figure 1 item 101 and column 10 lines 65-66)

executing initialization code of said debug agent, wherein replacing selected OS kernel code and data that are referenced, accessed, and otherwise used in the processing of debugging traps by the OS kernel, whereas said replaced code and data reside in or reference to said debug agent code and data images in memory. Madduri discloses when debug trap enable bit is set, hardware debug mode for all debug trap is entered instead of a software interrupt (see column 16 lines 17-23).

while the target is being debugged, the debug agent intercepting and processing one or more processor debugging traps generated. Madduri discloses entering hardware debug mode for all debug traps/faults (see column 16 lines 16-18).

restoring replaced OS kernel and data to original values. Madduri discloses an enable bit that causes entry into a hardware debug mode or software interrupt (see column 16 lines 17-19). When the enable bit is later unset, a software interrupt occurs during debug traps and faults, indicating restoring replaced OS kernel and data to original values.

However, Madduri fails to explicitly disclose:

when debugging is required, loading a debug agent from persistent store into memory and executing the debug agent.

when debugging is no longer required, unloading a debug agent.

Akgul discloses using debugger modules that are loaded into the target system memory and are executed only when they are needed for debugging purposes (see page 2 paragraph 18).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Madduri and Akgul to have debugger modules that are loaded into the target system memory and are executed only when they are needed for debugging purposes, indicating when debugging is required, loading a debug agent from persistent store into memory and executing the debug agent. A person of ordinary skill in the art would have been motivated to combine the teachings because Madduri is concerned with debugging a target system (see figure 1 and column 4 lines 32-40) and debugger modules that are loaded into the target system memory and are executed only when they are needed for debugging purpose, as per teaching of Akgul, provides debugging of embedded software without consuming significant additional amounts of memory of the target system and without requiring any dedicated debugging hardware and yet provides detailed and timely debugging information about the internals of the target platforms (see page 2 paragraph 16).

In regards to claim 16, Madduri discloses an apparatus, comprising:

a target computer, comprising one or more processors (see figure 1 item 101,104 and column 4 lines 32-35).

a memory coupled to the processor (see figure 1 item 106 and column 4 lines 32-35).

a hardware bus coupling the processor and one or more peripheral devices (see figure 1 item 106 and column 4 lines 32-35).

one or more communicating peripheral devices coupled to the hardware bus (see figure 1 item 106 and column 4 lines 50-52).

an operating system running on the processor (see figure 1 item 101 and column 10 lines 50-52).

one or more programs, each residing in memory and executing on the processor as one or more processes or threads and one or more device driver drivers, each loaded by the OS on demand. Madduri disclose the target system connecting to the host system (see figure 1 item 111 and 101 and column 4 lines 32-50). Thus, there inherently must be device drivers in order for the two systems to communicate.

a host computer, connecting to the target computer via communicating peripheral devices (see figure 1 item 111,112 and column 4 lines 46-50).

a host debugger executing on the host computer (see figure 1 item 111,112 and column 4 lines 46-50).

said debug agent initialization code replacing selected OS kernel code and data that are referenced, accessed, and otherwise used in the processing of debugging traps by the OS kernel, and whereas said replaced code and data reside in or reference to said debug agent code and data images in memory. Madduri discloses when debug trap enable bit is set, hardware debug mode for all debug trap is entered instead of a software interrupt (see column 16 lines 17-23).

while the target is under debug, said debug agent intercepting and processing one or more processor debugging traps generated. Madduri discloses entering hardware debug mode for all debug traps/faults (see column 16 lines 16-18).

restoring replaced OS kernel and data to original values when debugging is no longer required. adduri discloses an enable bit that causes entry into a hardware debug mode or software interrupt (see column 16 lines 17-19).

However, Madduri fails to explicitly disclose:

a debug agent, loaded by the OS on demand, residing in memory and executing on the target computer, wherein:

said debug agent is loaded from persistent store into memory and executed when debugging is required;

unloading the debug agent when debugging is no longer required

Akgul discloses using debugger modules that are loaded into the target system memory and are executed only when they are needed for debugging purposes (see page 2 paragraph 18).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Madduri and Akgul to have debugger modules that are loaded into the target system memory and are executed only when they are needed for debugging purposes, indicating a debug agent, loaded by the OS on demand, residing in memory and executing on the target computer, wherein said debug agent is loaded from persistent store into memory and executed when debugging is required and unloading the debug agent when debugging is no longer required. A person of ordinary skill in the art would have been motivated to combine the teachings because Madduri is concerned with debugging a target system (see figure 1 and column 4 lines 32-40) and debugger modules that are loaded into the target system memory and are executed only when they are needed for debugging purpose, as per teaching of Akgul, provides debugging of embedded software without consuming significant additional

amounts of memory of the target system and without requiring any dedicated debugging hardware and yet provides detailed and timely debugging information about the internals of the target platforms (see page 2 paragraph 16).

In regards to claim 22, Madduri in view of Akgul discloses the limitations as discussed above. Madduri further discloses

saving a entry in the OS syscall table pointing to a sys_init_module function, which services the OS module loading system call. Madduri discloses setting a bit that causes entry into a debug mode or software interrupt (OS module loading system call) (see column 16 lines 16-17). In order to toggle control back and forth between software interrupt and debug mode, an entry must be saved.

replacing said entry with a pointer to a proxy sys_init_module function residing in the debug agent memory image. Madduri discloses setting a bit that causes entry into a debug mode that otherwise cause a software interrupt (see column 16 lines 16-17).

responsive to system call to load the loadable module by the loading utility program, executing the steps comprising:

invoking said proxy sys_int_module function via said replacement. Madduri discloses setting a bit, indicating replacement, that causes entry into a debug mode (see column 16 lines 16-17).

responsive to determining that said module has been selected for debugging, initiating debugging of the loadable module by the said proxy sys_init_module function. Madduri discloses entry into a debug mode for all debug traps/fault of the processor core (see column 16 lines 16-17).

Claims 6, 21, and 36 rejected under 35 U.S.C. 103(a) as being unpatentable over Madduri in view of Akgul and in further view of US Patent 6,567,910 of Tessarolo et al. referred hereinafter "Tessarolo".

In regards to claims 6, 21, and 36, Madduri in view of Akgul discloses the limitations as discussed above. Madduri in view of Akgul fails to discloses wherein said debugging traps are generated by an event from the selected one of:

executing a processor BREAK instruction;

executing a invalid instruction;

executing an instruction causing data access failure;

generating any device interrupts or processor traps causing the processor to enter exception context.

However, Tessarolo discloses invalid instruction is one common cause of a debug trap (see column 30 lines 31-35).

It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of Madduri, Akgul, and Tessarolo such that said debug trap is generated by an invalid instruction. A person of ordinary skill in the art would have been motivated to combine the teachings because Madduri discloses processing debug traps (see page 16 lines 16-18), and invalid instructions, as per teachings of Tessarolo, constitute a known and common cause of a debug trap (see column 30 lines 31-35).

Allowable Subject Matter

Claims 2-5, 8-11,16,18-20,23-30,32-35, and 38-41 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action and to include all of the limitations of the base claim and any intervening claims.

Claims 12-15 would be allowable if rewritten or amended to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action and claims 42-45 would be allowable if rewritten or amended to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, and 35 U.S.C. 101 set forth in this Office action.

The reason for allowance for claims 12 and 42 is the inclusion of responsive to a run-control request, performing the steps comprising: setting a global variable to a value denoting transference of command loop to trap handler, whereas such variable is accessible to debug agent command loop and debug agent trap handler and responsive to determining that both the value of the global variable and the break code or the illegal instruction opcode denotes transference, the debug agent trap handler resuming execution to the debugged entity, comprising the steps: restoring original execution context of the debugged entity to the context saved area and executing the exception return code, resuming system execution to the destination at or near the point of the debugging trap occurrence in the debugged entity in conjunction with the rest of the limitation set forth in the claim.

The remaining claims, not specifically mentioned, are allowed because they are dependent upon one of the claim above.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

See Form PTO-892.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Emerson C. Puente whose telephone number is (571) 272-3652. The examiner can normally be reached on 8-5 M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert W. Beausoliel can be reached on (571) 272-3645. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Emerson Puente

Emerson Puente
Examiner
AU 2113